

**UNITED STATES PATENT APPLICATION FOR
SYSTEMS AND METHODS FOR LIGHTWEIGHT CONVERSATIONS**

Inventor:

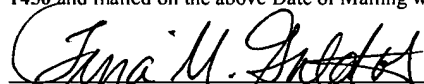
Chris Fry

**CERTIFICATE OF MAILING BY "EXPRESS MAIL"
UNDER 37 C.F.R. ' 1.10**

"Express Mail" mailing label number: EV 327619348 US

Date of Mailing: 12/11/, 2003

I hereby certify that this correspondence is being deposited with the United States Postal Service, utilizing the "Express Mail Post Office to Addressee" service addressed to **Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450** and mailed on the above Date of Mailing with the above "Express Mail" mailing label number.



Tina M. Galdos

Signature Date: 12/11/, 2003

SYSTEMS AND METHODS FOR LIGHTWEIGHT CONVERSATIONS

Inventor: Chris Fry

COPYRIGHT NOTICE

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document of the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

CLAIM OF PRIORITY

[0002] This application claims priority from the following application, which is hereby incorporated by reference in its entirety:

[0003] U.S. Provisional Application No. 60/450,139, entitled SYSTEMS AND METHODS FOR LIGHTWEIGHT CONVERSATIONS, by Chris Fry, filed on February 25, 2003 (Attorney Docket No. BEAS-01333US0).

CROSS-REFERENCED CASES

[0004] The following application is cross-referenced and incorporated herein by reference:

[0005] U.S. Patent Application No. 09/785,693, entitled CONVERSATION MANAGEMENT SYSTEM FOR ENTERPRISE WIDE ELECTRONIC COLLABORATION, by Rocky Stewart et al., filed February 16, 2001 (Attorney Docket No. BEAS-01033US5).

FIELD OF THE INVENTION

[0006] The present invention relates to the management of conversations in a cluster, such as for Web services.

BACKGROUND

[0007] Within the context of a business application, the collective set of well-defined and ordered messages exchanged between participants can be referred to as a "conversation". To manage these unique conversations, a typical approach utilizes a collaborative system that provides the capability to handle the conversation states, which may include, but are not limited to, programs, applications, services, database instances, etc.. As conversations are initiated, executed, and completed, a conversation manager can track and manage them, ensuring that the conversations are completed, and orchestrating the overall process

execution. Some of these collaborate systems maintain conversation state information in a database. Each time a request comes into a conversation space, the conversation manager must access a database containing the state information and load it for the particular conversation. This results in a resource-costly hit to a database each time a message is received in the conversation space.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figure 1 is a diagram showing a conversation system that can be used in accordance with one embodiment of the present invention.

[0009] Figure 2 is a diagram showing a failure scenario for the system of Figure 1 in an embodiment.

[0010] Figure 3 is a diagram showing another failure scenario for the system of Figure 1 in an embodiment.

[0010] Figure 4 is a diagram showing yet another failure scenario for the system of Figure 1 in an embodiment.

[0011] Figure 5 is a code example for an API that can be used with the system of Figure 1 in an embodiment.

DETAILED DESCRIPTION

[0012] The invention is illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to “an” or “one” embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0013] Systems and methods in accordance with one embodiment of the present invention overcome deficiencies in existing transaction management systems by taking advantage of a lightweight approach to conversation management. Such a lightweight approach can provide a significant increase in performance, which can be desirable when implementing and utilizing Web services, for example. A conversation can be setup between two processes, or conversation partners, each residing on a separate computing device that is capable of communicating through, by way of a non-limiting example, at least one public (e.g., internet) or private network (e.g., intranet), distributed system, client/server system, shared memory, etc.. One of the conversation partners might reside on a computing device in a computing device cluster, which might include a number of such devices. The computing device can be, but is not limited to, a mainframe computer, desktop or laptop PC, workstation, PDA, cell phone, electronic messaging device etc. on either the client or the server side of an application. When a message for a conversation comes into that computing device cluster, an identifier (ID), which, by way of a non-limiting example, can be a string of letters and digits, will be included with that message. When this message

comes into one of the computing devices in the cluster, it is necessary to determine whether the cluster is maintaining the state correlated with that ID. If the cluster is maintaining such state, it is necessary to determine the location of the state.

[0014] In a cluster, there can be a conversation manager that keeps track of state location throughout the cluster. The conversation manager can be, for example, a management object or a management service. When a computing device receives a request, that computing device can contact the conversation manager to determine the location of the state within the cluster. An example of this is shown in **Figure 1**. In one embodiment, a conversation partner **100** sends a message on the conversation that is received by a computing device **106** in the cluster **102**. That computing device **106** can contact the conversation manager **104** to determine the location of the state for the message containing a particular conversation ID. The conversation manager can inform computing device **106** that the state for that conversation ID is located on computing device **108**. Computing device **106** can then route to that device, or directly invoke the state from computing device **108**. In one embodiment and by way of non-limiting examples, state can be invoked over a Java Remote Method Invocation (RMI), a Common Object Request Broker Architecture (CORBA), or any other application service. Approaches for utilizing these types of services are well known to those of ordinary skill in the art. An exemplary API that can be used with the system of **Figure 1** is shown in **Figure 5**.

[0015] In one embodiment, each computing device in a cluster can maintain an internal map of every state leased (i.e., access licensed to a client for a certain period of time), owned, or stored on it. Such state information can be stored in-memory on computing devices in the cluster. This gives the conversation management its “lightweight” characteristics. Since all state information is stored in-memory, there is no need to do a resource-intensive database lookup for each message received. This can provide a significant increase in performance. The increase in performance can be balanced with the need for reliability, as keeping everything in-memory can decrease reliability as a cluster failure can cause all state information to be lost.

[0016] In one embodiment, if a computing device receives a request for which that computing device owns the appropriate state, the computing device can simply answer the request directly without contacting the conversation manager. This in and of itself can be a substantial performance enhancement over existing systems. If the computing device has to do a “lookup” by contacting the conversation manager, the computing device can use the location information to do an RMI lookup instead of a Web service lookup, which can also provide a performance benefit.

[0017] Certain embodiments can allow a user to determine the balance between reliability and performance. For example, reliability can be improved by storing at least some state information to persistent storage, which may include, but is not limited to, database, hard drive, and other mass storage

device. These embodiments can improve reliability, but may decrease performance and increase the number of conditions to be satisfied and monitored throughout the cluster.

[0018] If a computing device in a conversation cluster becomes unavailable, the conversation manager may not have any indication that the computing device is not available. The conversation manager can still inform a computing device receiving the request of the location stored by the conversation, and it is only when the receiving computing device attempts to invoke a state on that failed computing device that the receiving computing device learns that the failed computing device is unavailable. In certain embodiments, it is possible for a conversation manager to periodically check for the availability of cluster computing devices, such as by periodically pinging them.

[0019] A conversation manager can be useful in situations where at least two clients are sharing the same conversational state. In one embodiment, a centralized conversation manager can create new conversations and controls access to conversations. The session state of a Web service conversation can offer primary and secondary replication with primaries and secondaries being distributed across the cluster. A conversation manager can be capable of replication, and capable of recovering from failure. Further, when a primary computing device fails, the conversation manager can be capable of routing to the secondary computing device, such that the secondary computing device becomes the new primary and a new secondary is created.

[0020] In one embodiment, a conversation manager can control the location and leasing of a conversational state within a cluster. The conversation manager can consist of a hash table lookup of conversation IDs to states' locations. The conversation manager can grant leases to the various states. If a state has a lease on the conversation, it can freely handle any incoming invoke requests. If an invoking request is received and the lease for a state has expired, the state can attempt to gain a new lease, or lease extension, from the conversation manager. If the lease is already granted to another computing device, the request can be redirected to the new location of the state, and the current instance can be destroyed.

[0021] Conversations may or may not be available to maintain shared state. If conversations can share state, multiple Web services can be joined using the shared state during their sessions. If conversations cannot share state, a conversation manager can be used to maintain the state information in a central location, and information about the completion of the participants can be coordinated the same place where the Web service is offered, and to which requests are sent for executing it. A conversation manager can be used to track participating Web services that register to join a state, and managers themselves can become participants that represent other Web services.

[0022] In one exemplary failure scenario, a primary can fail without a subordinate conversation. In this case, as shown in **Figure 2**, the primary (B) fails, the conversation manager routes client A to the secondary (C), which becomes the new primary and (D) becomes the new secondary. In a second failure

scenario where the primary (B) fails with subordinate conversation, as shown in **Figure 3**, clients A & E are both routed to the new primary (C). In a third failure scenario where the primary (B) freezes, as shown in **Figure 4**, primary (B) intermittently fails. The conversation manager routes client E to the new primary (C) and cleans up the old primary.

[0023] One embodiment may be implemented using a conventional general purpose or a specialized digital computer or microprocessor(s) programmed according to the teachings of the present disclosure, as will be apparent to those skilled in the computer art. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art. The invention may also be implemented by the preparation of integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

[0024] One embodiment includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the features presented herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, micro drive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[0025] Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, execution environments/containers, and applications.

[0026] The foregoing description of the preferred embodiments of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. Embodiments were chosen and described in order to best describe the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention, the various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.